Manufacturing Resource Planning (MRP) using Google Colab

Table of Contents

Course Objective

This course provides a **hands-on approach** to applying **Manufacturing Resource Planning (MRP)** using Python in **Google Colab**. MRP helps businesses efficiently manage **materials, production schedules, and inventory** to optimize manufacturing processes.

By the end of this course, you will:

- Understand the fundamentals of **MRP and Production Planning**
- Learn how to use **Python for demand forecasting, material tracking, and inventory optimization**
- Build a **Bill of Materials (BOM) system** for manufacturing workflows
- Generate **Production Schedules** based on lead times and material availability

Introduction to MRP

Why MRP in Manufacturing?

- The importance of **materials management** in production
- Role of **MRP in reducing waste and improving efficiency**
- Overview of **Python for data-driven manufacturing**

Environment Setup

Ensure you have **Google Colab** or a Python 3.x environment with the following:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Creating MRP Data

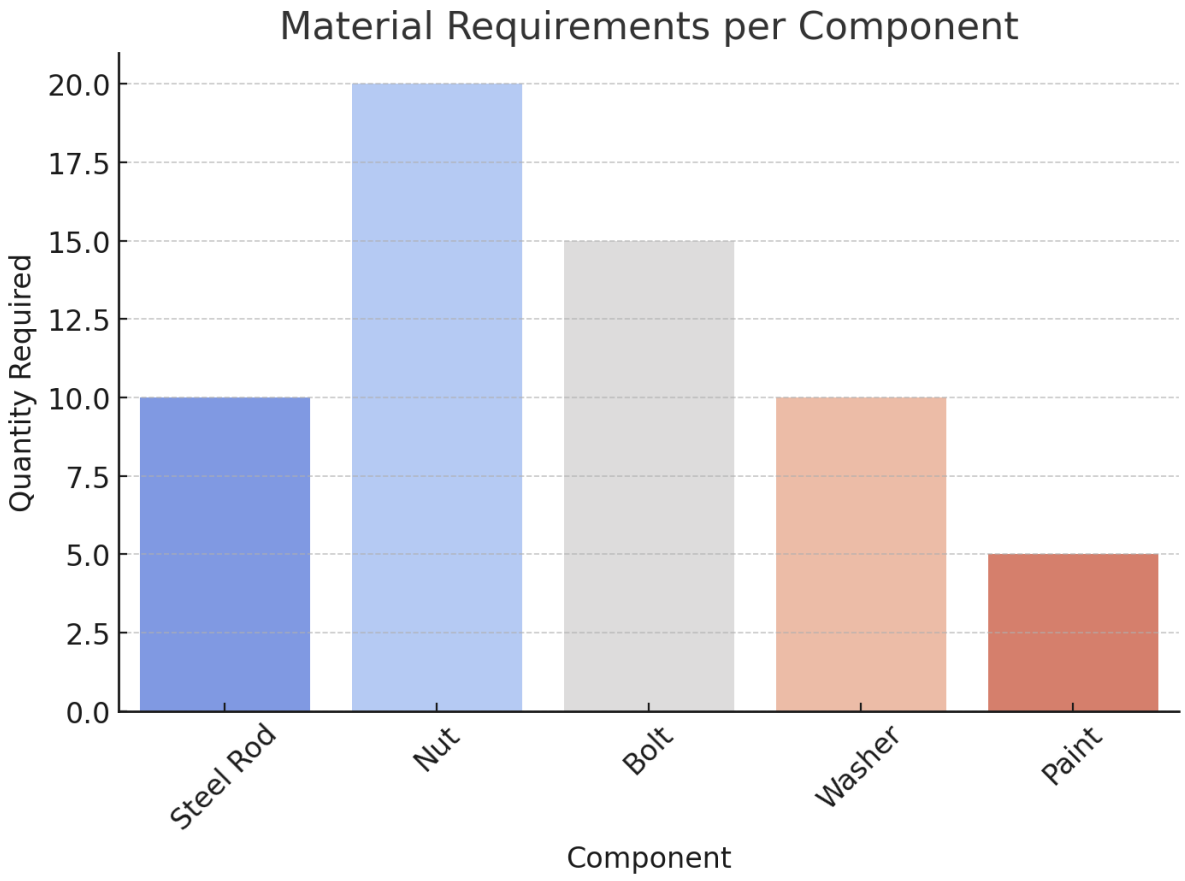
Sample Dataset: Bill of Materials (BOM)

```
data = {
    'Component': ['Steel Rod', 'Nut', 'Bolt', 'Washer',
'Paint'],
    'Lead Time (Days)': [5, 2, 3, 1, 4],
    'Quantity Required': [10, 20, 15, 10, 5]
}

bom_df = pd.DataFrame(data)
bom_df
```

Data Visualization: Material Requirements

```
plt.figure(figsize=(8,5))
sns.barplot(x='Component', y='Quantity Required', data=bom_df,
palette='coolwarm')
plt.title('Material Requirements per Component')
plt.xlabel('Component')
plt.ylabel('Quantity Required')
plt.xticks(rotation=45)
plt.show()
```



Demand Forecasting

Generating Sample Demand Data

```
np.random.seed(42)
dates = pd.date_range(start='2023-01-01', periods=60, freq='D')
demand = np.random.randint(50, 200, size=60)

demand_df = pd.DataFrame({'Date': dates, 'Demand': demand})
demand_df.set_index('Date', inplace=True)
demand_df.head()
```

Visualizing Demand Trends

```
plt.figure(figsize=(10,5))
sns.lineplot(x=demand_df.index, y=demand_df['Demand'],
color='blue', label='Daily Demand')
plt.title('Manufacturing Demand Trend')
plt.xlabel('Date')
plt.ylabel('Demand')
plt.legend()
plt.grid()
plt.show()
```

Production Scheduling

Computing Production Lead Time

```
bom_df['Total Lead Time'] = bom_df['Lead Time (Days)'] *
bom_df['Quantity Required']
bom_df
```

Generating a Production Schedule

```
schedule = {
    'Product': ['Product A', 'Product B', 'Product C'],
    'Start Date': ['2023-04-01', '2023-04-05', '2023-04-10'],
    'End Date': ['2023-04-15', '2023-04-20', '2023-04-30'],
    'Quantity': [500, 750, 1000]
}
schedule_df = pd.DataFrame(schedule)
schedule_df
```

Visualizing Production Timeline

```
plt.figure(figsize=(8,5))
sns.barplot(x='Product', y='Quantity', data=schedule_df,
palette='viridis')
plt.title('Production Plan')
plt.xlabel('Product')
plt.ylabel('Quantity to be Produced')
plt.show()
```

Inventory Optimization

Calculating Safety Stock

```
lead_time_demand = demand_df['Demand'].rolling(window=7).mean()
safety_stock = lead_time_demand.mean() * 1.5  # 1.5x buffer
print(f"Optimal Safety Stock: {safety_stock:.2f} units")
```

Visualizing Inventory Levels

```
plt.figure(figsize=(10,5))
sns.histplot(demand_df['Demand'], bins=15, kde=True,
color='green')
plt.axvline(safety_stock, color='red', linestyle='--',
label='Safety Stock Level')
plt.title('Inventory Demand Distribution')
plt.xlabel('Demand Level')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```

Course Summary

- Applied MRP concepts in Python for Manufacturing Planning
- Generated and Visualized Demand Data for Forecasting

- Developed a Bill of Materials (BOM) with Lead Time Calculations
- Created Production Schedules and Optimized Inventory Levels